BURGEVIN Valentin GR IMA

DURET Guillaume

TP Equation aux Dérivées Partielles :

2018-2019

# INTRODUCTION :

Le but de ce travail est de pouvoir débruiter une image, pour cela nous allons appliquer les équations différentielles partielles.

Nous allons utiliser un filtre gaussien en 2D pour lisser l’image bruitée et nous allons améliorer cette méthode à l’aide de coefficients de diffusion variant en fonction des gradients de l’image.

# METHODE :

Pour réduire le bruit d’une image on applique un filtre gaussien tel que l’on obtienne l’image :

Avec : le filtre gaussien en 2D avec

Et l’image que l’on veut débruiter

De plus on utilise une convolution en dimension 2 qui correspond à :

De plus on peut montrer que pour qui a été convolué par un filtre gaussien décrit précédemment vérifie toujours l’équation de la chaleur :

La variable t joue le rôle du temps et fait varier la largeur du filtre gaussien et donc de plus en lisser l’image.

De plus du fait que l’on veut appliquer cette équation on discrétise ces équations en utilisant les taux de variation dans les 4 directions de l’image.

South : -

North : -

East : -

West : -

On a donc les discrétisations suivantes

Et

En remplaçant dans l’équation de la chaleur on obtient :

Ce qui nous permet d’obtenir finalement :

De plus on pose au début de l’algorithme :

On verra que cette solution permet bien de réduire le bruit d’une image cependant celle-ci lisse aussi les contours de l’image ce qui rend l’image floue.

Pour resoudre ce problème on ajoute un facteur de Diffusion pour pouvoir controler le lissage selon les positions sur l’image. En effet le but est d’appliquer un coefficient D qui permet de lisser les zones homogènes mais pas les de fort gradient(contours) c’est-à-dire une fonction décroissante vérifiant :

(stop fort gradient)

(lissage faible gradient)

Or

Tout comme précédemment on discrétise et montre que

function I=make\_your\_image(n)

w1=floor(n/3);

w2=floor(n/2);

bw=floor(n/50);

vp=255;

imagemat=vp\*ones(n,n);

imagemat(:,w2-bw:w2+bw)=0;

imagemat(w1-bw:w1+bw,:)=0;

imagemat(2\*w1-bw:2\*w1+bw,:)=0;

I=double(imagemat);

t\_final=4; %temps final

tau=0.2; %Pas du faux temps

n=300; %taille de l'image

imag1

%croix bizarre

u=make\_your\_image(n);

%lena

% u=imread('Lena.jpg');

% u=im2double(rgb2gray(u));

% Joyeux Anniversaire

% u=im2double(imread('JoyeuxAnniversaire.gif'));

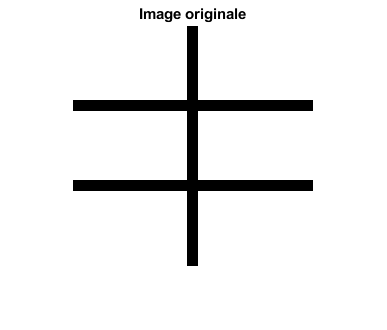
[w,h]=size(u);

figure(1)

subplot(221)

imshow(u,[])

title('Image originale')



bruit

% b=wgn(w,h,30);

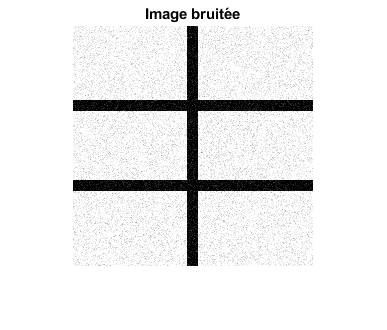
% I\_bruit=I-b;

u0=imnoise(u,'gaussian');

subplot(222)

imshow(u0,[])

title('Image bruitÃ©e')



debruitage 1

u\_debruite=u0;

for m=1:tau:t\_final

for i=2:w-1

for j=2:h-1

derive\_sud=u\_debruite(i+1,j)-u\_debruite(i,j);

derive\_nord=u\_debruite(i-1,j)-u\_debruite(i,j);

derive\_ouest=u\_debruite(i,j+1)-u\_debruite(i,j);

derive\_est=u\_debruite(i,j-1)-u\_debruite(i,j);

u\_debruite(i,j)=u\_debruite(i,j)+tau.\*(derive\_sud+derive\_nord+derive\_ouest+derive\_est);

end

end

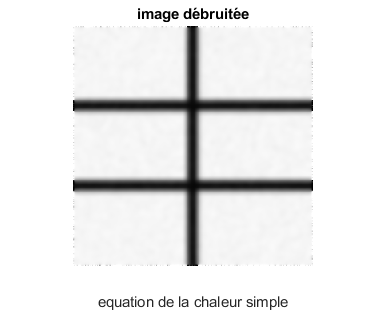
end

subplot(234)

imshow(u\_debruite,[])

title(['image dÃ©bruitÃ©e pour alpha =',num2str(alpha) ])

xlabel('equation de la chaleur simple')



gauss

alpha = 0.2;

D=@(x) exp(-x.^2./(2\*alpha^2));

u\_gauss=u0;

for m=1:tau:t\_final %temps

for i=2:w-1 %ligne

for j=2:h-1 %colonne

derive\_sud=u\_gauss(i+1,j)-u\_gauss(i,j);

derive\_nord=u\_gauss(i-1,j)-u\_gauss(i,j);

derive\_ouest=u\_gauss(i,j+1)-u\_gauss(i,j);

derive\_est=u\_gauss(i,j-1)-u\_gauss(i,j);

u\_gauss(i,j)=u\_gauss(i,j)+tau\*(D(derive\_sud)\*derive\_sud+D(derive\_nord)\*derive\_nord+D(derive\_ouest)\*derive\_ouest+D(derive\_est)\*derive\_est);

end

end

end

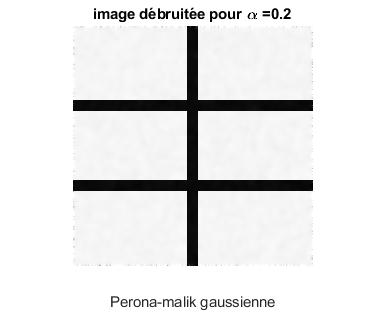
subplot(235)

imshow(u\_gauss,[])

title(['image dÃ©bruitÃ©e pour \alpha =',num2str(alpha) ])

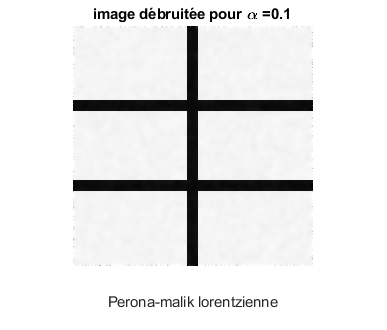
xlabel('Perona-malik gaussienne')

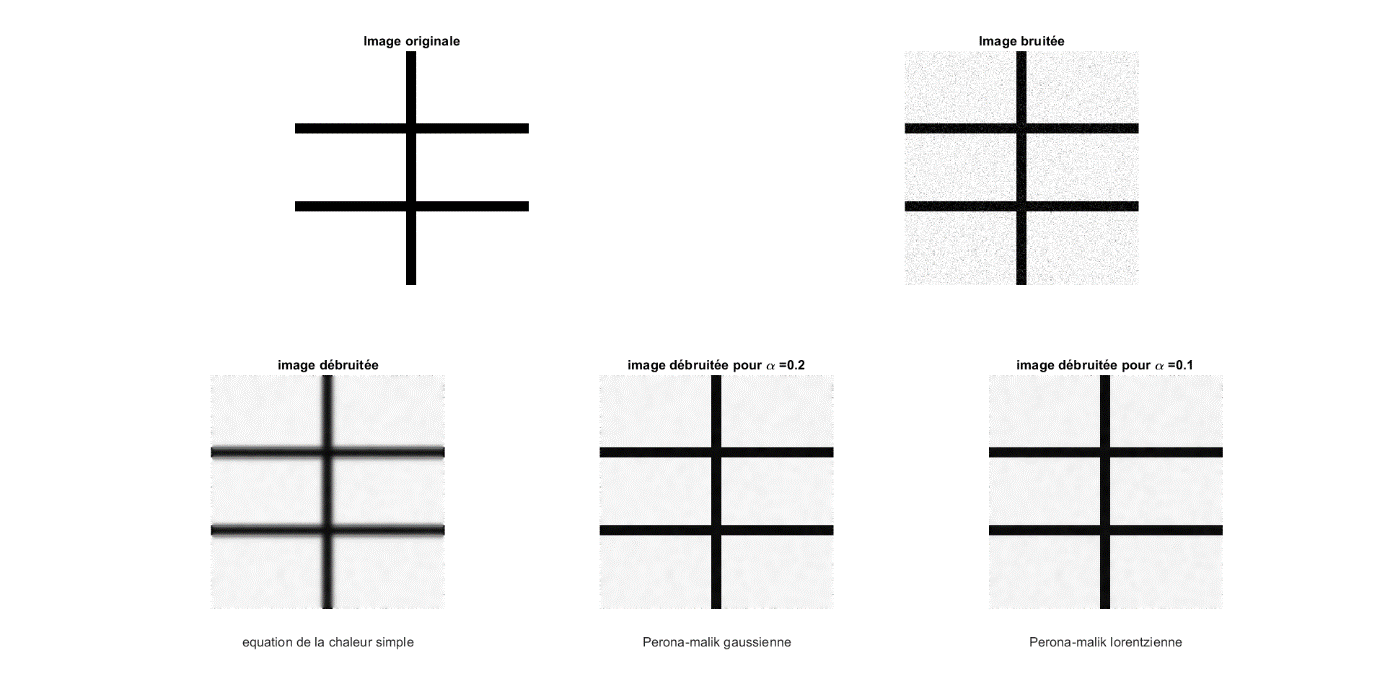
loren



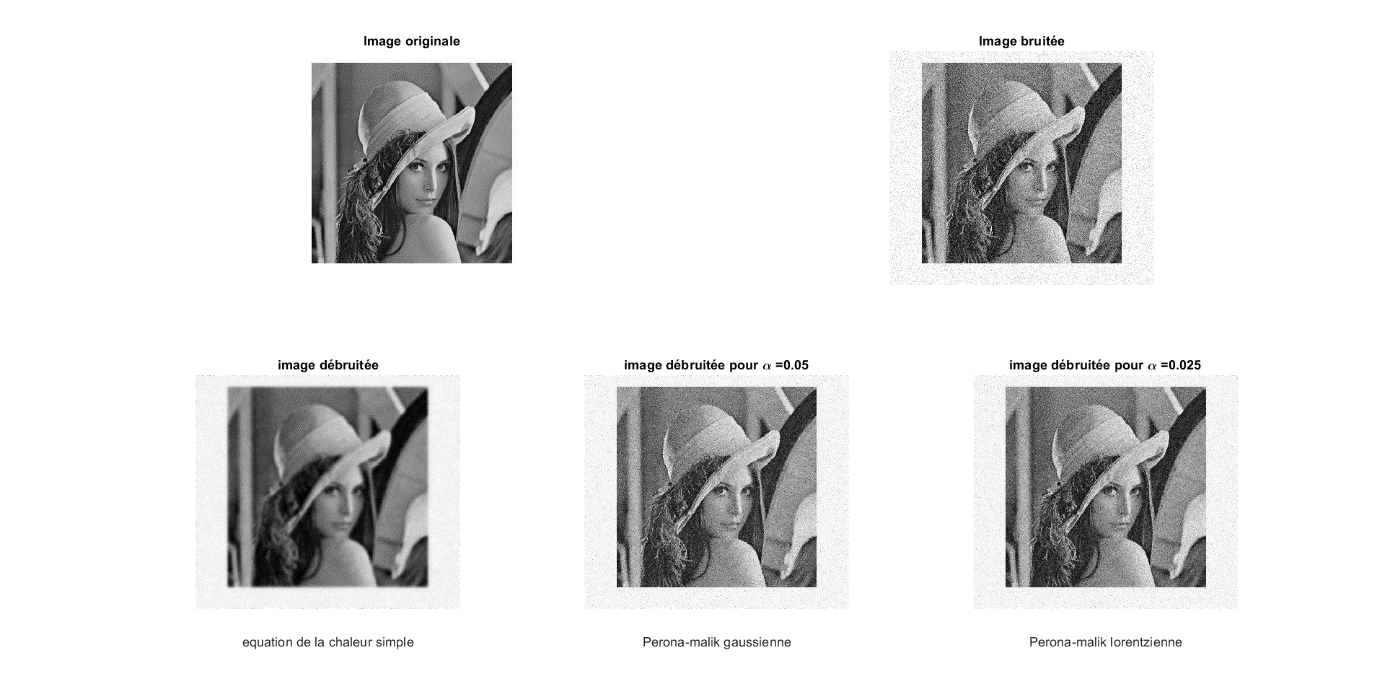
alpha = 0.1;

D=@(x) 1/(1+(x.^2./alpha^2));

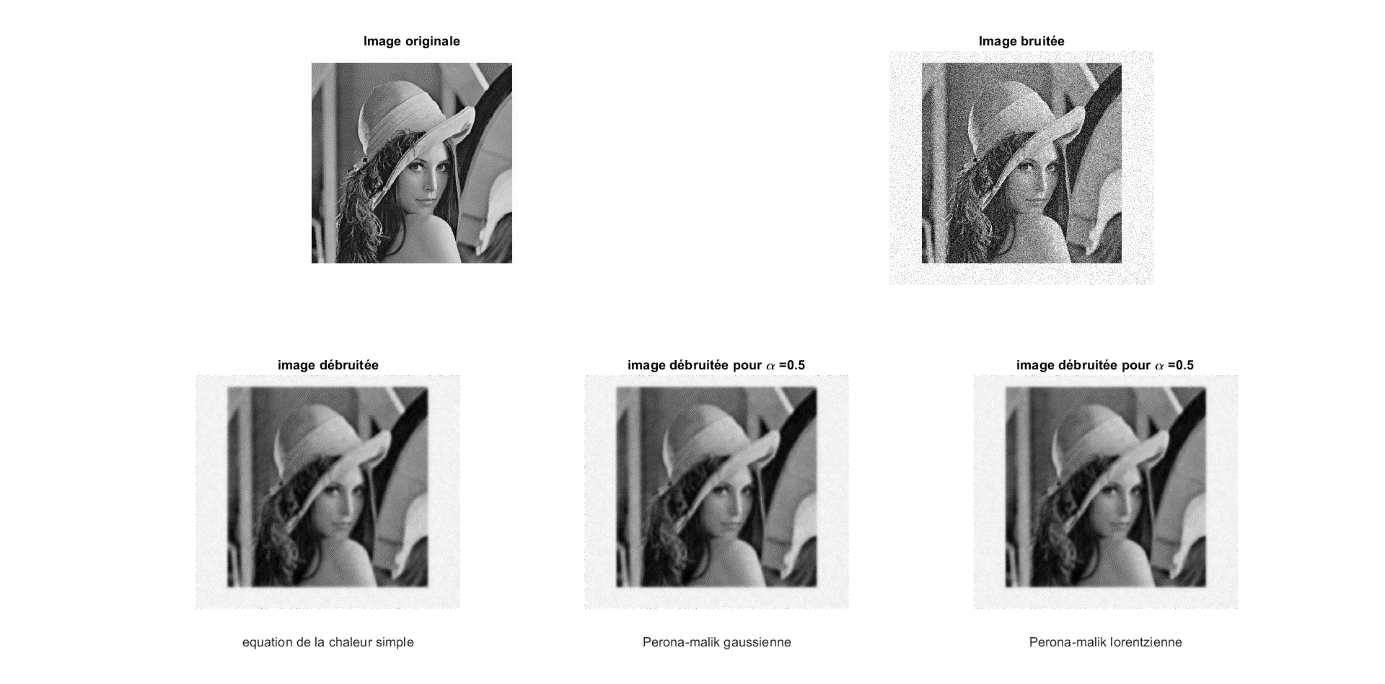




Alpha petit



Alpha grand



Alpha bien

